



RECEIVED
GROUP 230

24

1988 AUG -9 PM 9:29

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

APPLICATION OF)

MARTIN G. REIFFIN)

FILED: APRIL 3, 1985)

FOR: COMPUTER SYSTEM WITH)
REAL-TIME CODE PROCESSING)

SERIAL NO. 719,507)

ART UNIT 232

Examiner T. Lee

THIRD AFFIDAVIT of MARTIN G. REIFFIN

County of Contra Costa)
State of California) ss.

MARTIN G. REIFFIN, being duly sworn, deposes and says
that:

1. I am the applicant and inventor in the above-
entitled application and the affiant of the previously filed
Reiffin affidavit executed November 24, 1986.

2. I hold the degrees of Bachelor of Mechanical
Engineering, Master of Science in Electrical Engineering, and
Juris Doctor. I am admitted to the bar in the States of
Michigan, Illinois and New York. I have taken courses in
computer science at the University of California at Irvine,
including the two compiler courses described in the first Mark
Wadsworth affidavit filed in applicant's prior copending
application Serial No. 425,612.

3. I am a former examiner in the United States Patent

Office. I have been admitted to practise before the Office since 1948. I have practised patent law from that time until 1979.

4. For fifteen (15) years I represented International Business Machines Corporation in computer-related litigation in the Federal Courts in the Second, Third, Fifth and Ninth Circuits, in computer-related interferences in the United States Patent Office and before the Court of Customs and Patent Appeals, and in the preparation and prosecution of computer-related patent applications.

5. Since 1980 I have devoted full time to the invention of circuits and compilers. During this time my circuit patents were licensed to nineteen (19) licensees who have paid me about three million dollars (\$3,000,000.00) in royalties for said licenses.

6. The Lawrence et al. patent No. 4,464,730 does not contain an enabling disclosure of either software or hardware for performing the functions attributed to it by the Examiner, for the reasons stated below.

7. The "interpreter/formatter" of said Lawrence is disclosed in the patent drawings solely as follows:

In Fig. 1 as a rectangular box 6 bearing the legend "I/F";

In Fig. 3 as a rectangular box 6 bearing the legend "INTERPRETER/FORMATTER";

In Fig. 4 as a dashed-line box 6 bearing the legend "I/F" and having two dashed-lines with arrowheads

extending therefrom; and

In Fig. 5 as a segment 38 of a box 33 and bearing the legend "(TEXT CODE)", and described in the specification (Col. 6, Lines 62-64) as containing "code for controlling the segmenting, formatting and editing processes 5, 11, 6 and 10, FIG. 1." No such code is disclosed in the specification, nor is there any hint as to where such code might be obtained.

8. Prior to the Examiner's citation of the Lawrence patent affiant had never seen or heard of the term "interpreter/formatter". Upon information and belief, the term "interpreter/formatter" appears to be a coined phrase which is not a term of art in the computer industry, and insofar as affiant is aware, this term is neither defined, explained nor disclosed in any dictionary, treatise, encyclopedia, other patent, or other publication. Upon information and belief, the term "interpreter/formatter" does not identify any particular program and does not suggest to one skilled in the art any other available program which may be substituted for it.

9. The portion "formatter" of the Lawrence term "interpreter/formatter", together with the names of functions allegedly attributed by the patent specification to this undisclosed and unknown program, suggests that the "interpreter/formatter" is a text or screen formatter which is purported to have several modes of interaction with the text editor 10. These modes are summarily and briefly alleged in the Lawrence specification as follows:

(1) the interpreter/formatter 6 is "initiated...by the text editor 10 on completion of an update to

the text in the store 2" (Col. 3, Lines 25-29);

(2) the interpreter/formatter 6 is first "invoked" by the text editor 10 "to determine the physical cursor position relative to the text in the store 2", and after an insertion or deletion is "re-invoked" by the text editor 10 "to display the effect of the change on the screen" (Col. 3, Lines 30-38); and

(3) the "interpreter" is "terminated at the end of the current row" if "a key is depressed during formatting" (Col. 11, Lines 51-59).

10. The mere depiction in the Lawrence patent drawings of a rectangular box labeled "I/F" or "INTERPRETER/FORMATTER" or "(TEXT CODE)" does not constitute an enabling disclosure of software or hardware, in that it does not teach, suggest or otherwise make obvious to one skilled in the computer art how to write software or build hardware to perform the above-designated interactions for "invoking" and "terminating" the interpreter/formatter 6, nor is there anything else in the Lawrence patent disclosure which might teach one how to create such software or hardware.

11. Applicant's Claim 51 recites:

"the code processor program thereafter continuously maintains control of the central processing unit, except during interrupts"; and

"means immediately responsive to activation of the interrupt input...thereby immediately passing control of the central processing unit to the interrupt service routine...[the latter] including an editor program".

These limitations are not descriptive of the Lawrence interpreter/formatter 6 and text editor 10. As explained below, the Lawrence interpreter/formatter does not continuously maintain control of the central processing unit, and the Lawrence text editor 10 is not disclosed as included in an interrupt service routine which immediately takes control upon activation of an interrupt.

12. In Paragraph 11 of the final rejection the Examiner attributes to the Lawrence et al. disclosure the following interactions:

"During the formatting operation, the interpreter/formatter normally has control of the system."

"Upon the activation of a keystroke, the interpreter/formatter is interrupted and the control of the system is passed to the editor."

"In other words, control of the system is passed back to the interpreter/formatter from the editor after each of the keystroke [sic]."

These alleged interactions are not descriptive of the Lawrence interpreter/formatter 6 and text editor 10, particularly if the terms "interrupt", "control" and "normally" are used in accordance with the regular and accepted usage in the computer art and with the usage in applicant's claims and specification. As explained below, the Lawrence interpreter/formatter does not normally or continuously maintain control of the central processing unit, and the Lawrence text editor 10 is not disclosed as included in an interrupt service routine which immediately takes control upon activation of an interrupt.

13. The fact that the Lawrence interpreter/formatter is described as "terminated" if a key is depressed during formatting is inconsistent with the Examiner's contention that the

interpreter/formatter has been "interrupted" or that the interpreter/formatter continuously maintains control "except during interrupts". The terms "terminated" and "interrupted" have different meanings in the computer art. A routine is said to be "terminated" when it has either completed its sequence of instructions or has aborted. In either event, the routine will not execute again unless and until it is invoked all over again from the beginning of its instruction sequence. On the other hand, a routine is said to be "interrupted" when it temporarily loses control of the central processing unit to an interrupt service routine. When the latter is finished, the interrupted routine then automatically resumes its control of the central processing unit and continues executing its instructions at the very point in the instruction sequence where it was interrupted.

14. At Col. 3, Lines 30-38 the Lawrence specification states that the text editor 10 "invokes" and "re-invokes" the interpreter/formatter 6. In the computer art a routine (procedure or subprogram) is said to be "invoked" by another routine when it is "called" by the latter. The first (calling) routine invokes the second (invoked) routine by executing a "call" instruction. This stores the memory address of the invoked routine into the program counter of the computer. The central processing unit then obeys the program counter and jumps to the address of the invoked routine by executing the instructions stored at said address. That is, the first (calling) routine knows and determines the address in memory to which the central processing unit is to jump. Such a call or

invocation is different from an interrupt. In the interrupt operation the first routine has no knowledge of the second routine and has no control of the destination of the central processing unit or of when the jump to the second routine takes place. Insofar as the first routine is concerned, all it "knows" is that it has temporarily lost control of the central processing unit to some unknown interrupt service routine and that it will automatically regain its control when the interrupt service routine has finished.

15. The fact that the Lawrence interpreter/formatter is described as being "invoked" and "re-invoked" is inconsistent with the Examiner's contention that it maintains control of the system except during interrupts. As explained above, an interrupted routine does not regain control by being "invoked". Instead it regains control automatically when the interrupt service routine terminates.

16. Furthermore, when a first routine is interrupted by a second routine, the latter gets control of the central processing unit immediately, and not after some other subsequent event such as the interrupt/formatter reaching the end of the current row of the text being processed. Col. 11, Lines 51-54 of the Lawrence specification states:

"If a key is depressed during formatting, then the interpreter is terminated at the end of the current row and control is passed back to the keystroke processor 9". (Emphasis added.)

If the keystroke processor 9 and text editor 10 were embodied in the interrupt service routine, they would get control immediately upon activation of the interrupt rather than waiting until after

the interpreter/formatter reaches "the end of the current row" of the text being formatted. This is inherent and inevitable by the very definition and nature of an interrupt.

17. The Examiner's statement in Paragraph 11 of the final rejection that:

"During the formatting operation, the interpreter/formatter normally has control of the system."

is either tautological or inaccurate. If taken literally, it is merely a tautology; that is, a statement which is inevitably true solely by virtue of its logical form. Equally true and meaningless would be a statement such as:

"During the segmenting operation, the segmenter normally has control of the system."

18. If the Examiner is trying to say that the interpreter/formatter normally has control of the system in the same sense that applicant's Claim 51 recites that his code processor program continuously maintains control, then the Examiner's statement is contrary to the express wording of the Lawrence et al. specification. The latter states that the interpreter/formatter gets control only when it is repeatedly "invoked" or "re-invoked" by the text editor 10 or segmenter 5.

19. The Lawrence specification (Col. 6, Lines 48-49) mentions "an interrupt line 29" shown in Fig. 5 as extending between the microprocessor 26 and keyboard adapter 30. This does not disclose or suggest the mode of operation recited in applicant's claims. At very most, this might suggest to one skilled in the art the conventional operation of the Personal Computer of IBM, the assignee of the Lawrence et al. patent. As shown in Exhibit A which includes copies of Pages 92-101 of "8088 Assembly Language Programming The IBM PC" by D. C. Willen and J.

L. Krantz, in the IBM Personal Computer the striking of a key activates an interrupt which stores a corresponding character code in a type-ahead buffer. The main program (e.g. the Lawrence interpreter/formatter 6) running at the time then immediately resumes control of the central processing unit. No successor routine (e.g. Lawrence's text editor 10) takes control as an interrupt service routine. To change control from the main program to an editor it would be necessary for the main program to terminate or to invoke the editor. The former would appear to be disclosed by Lawrence et al. Although it would be possible for a programmer to change the interrupt vectors stored in the IBM Personal Computer so that the interrupt service routine includes an editor which is directly and immediately invoked by the interrupt, this modification is not the conventional mode of operation and is not disclosed in the Lawrence et al. patent. There is nothing in their disclosure to suggest that activation of their interrupt line 29 does anymore than the conventional keystroke operation in the IBM Personal Computer.

20. The Examiner's statement that:

"... control of the system is passed back to the interpreter/formatter from the editor after each of the keystroke [sic]."

is unsupported by and inconsistent with the Lawrence disclosure. The Examiner erroneously reads this "interaction" into the claim limitation "keystroke-by-keystroke basis" in Column 13, Line 29. However, the context of this limitation reveals that the Examiner's interpretation is unfounded. This limitation refers only to the "editing means" and has no relation to the "formatting means" which is not recited until the last paragraph of the claim.

21. Furthermore, in this last paragraph it is recited that the formatting means is invoked "when a text editing step has been completed." Since a text editing step usually comprises a plurality of keystrokes, there is no disclosure in support of the Examiner's statement that control is passed back after each keystroke.

22. Furthermore, there would be no reason or purpose for passing control back to the interpreter/formatter after each keystroke and before the plurality of keystrokes comprising an editing operation is completed. A screen formatter, such as the Lawrence interpreter/formatter purports to be, takes only a fraction of a second to format a screenful of text and it would be pointless to reformat the entire screen repeatedly for each keystroke and thereby reformat the screen numerous times for each complete editing operation. On the other hand, in applicant's system a compilation may take a very long time and hence it is vital that processing be done between keystrokes to minimize the tedious waiting periods inherent in conventional compilation methods.

Martin G. Reiffin
Martin G. Reiffin

Subscribed and sworn to before me this 29th day of

July, 1988.



Diane M. Spagnoletti
Notary Public

